C2 - Python for Data Science

C-DAT-100

HTTP Requests

Microsoft Teams API

1.0

HTTP Requests

delivery method: py03 on Github
language: python

In data science (actually, more broadly in all computer science) communicating though an **API** to retreive DATA is quite common.
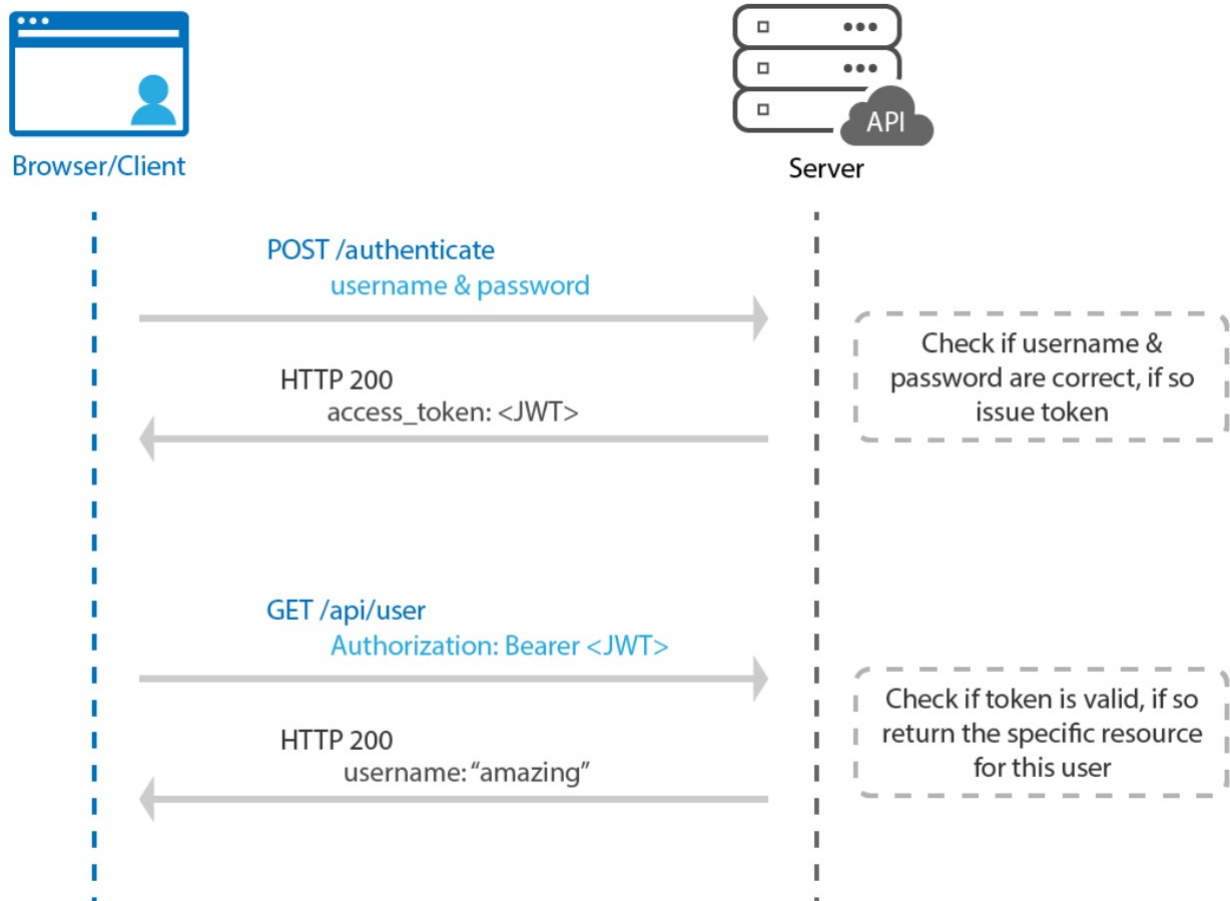

CommitStrip.com

Intro

Before you start, you need to get **your** Microsoft Team's **access token**.

Go to **https://developer.microsoft.com/en-us/graph/graph-explorer** and log in with your epitech student account.

You can copy your token from the access token tab.

Use this API endpoint : **https://graph.microsoft.com/beta**

This token has a lifespam of 1 hour. You need to refresh it every hour for communicating with Microsoft Teams API.
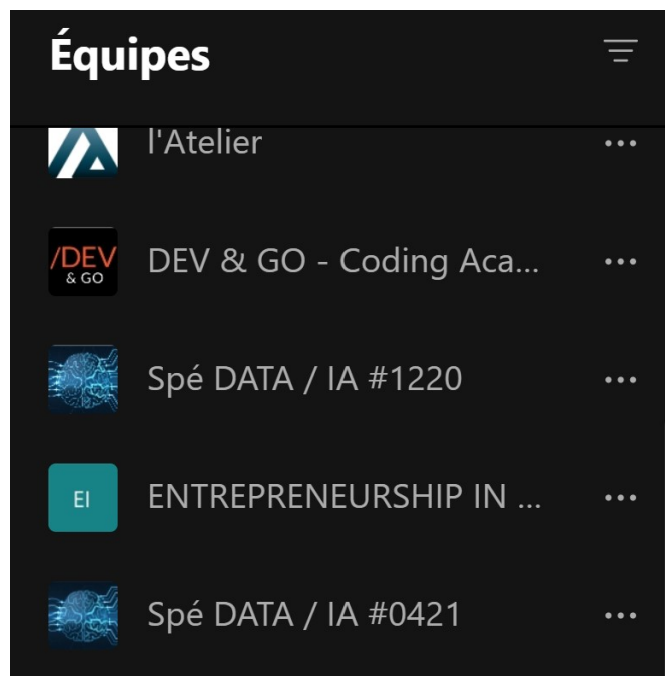
Exercise 01

Create a function `member_of` that returns the **display_name** and the **id** of all **team** you belong to.

Route : **/me/memberOf**

Exemple

```
teams = member_of()
for name, id in teams.items() :
    print(name)
    print(id)
```
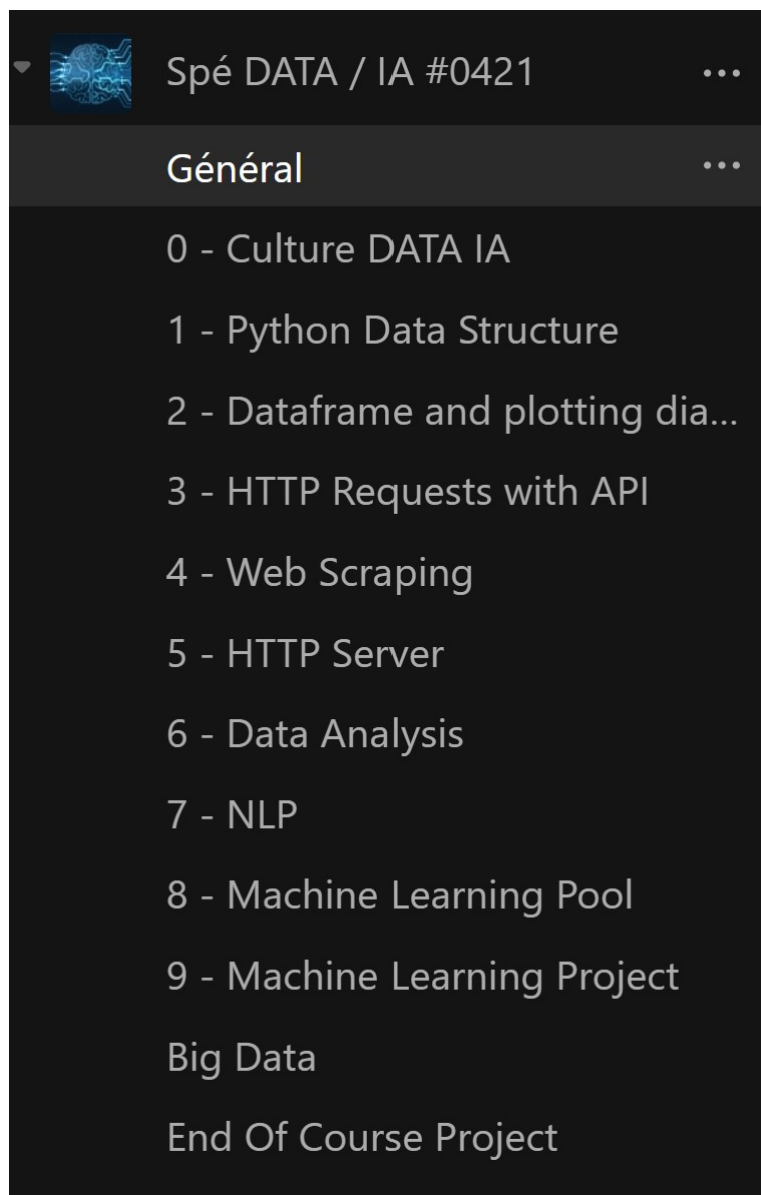
Exercise 02

Create a function `get_channels(team_id)` that returns all channels from the team_id given as parameter.

Route : **/teams/{team_id}/channels**

Exemple

```
channels = get_channels('c1995755-c035-4c13-bc82-6b65b5c1d206')
for channel, id in channels.items() :
    print(channel)
    print(id)
```
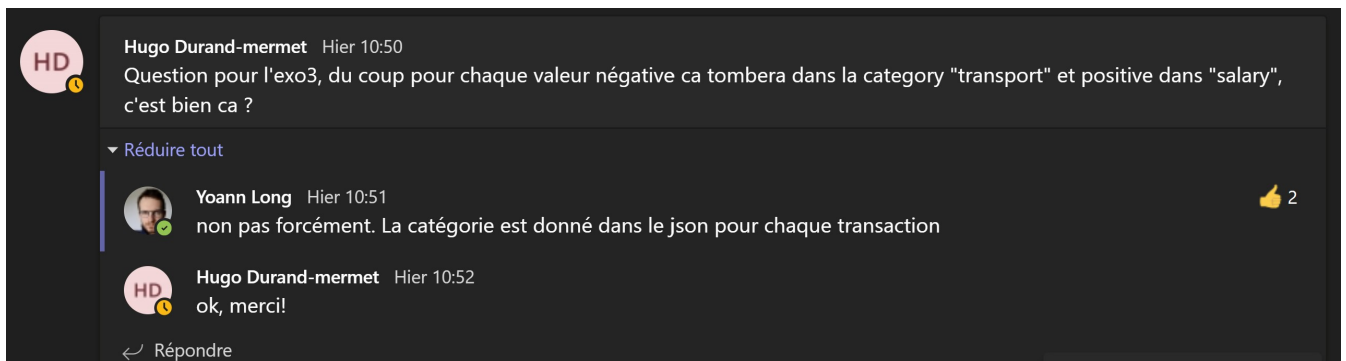
Exercise 03

Create a function `get_messages(team_id, channel_id)` that returns all messages and the replies from the channel_id of the team_id given as parameters.

Routes :
- **/teams/{team_id}/channels/{channel_id}/messages**

- **/teams/{team_id}/channels/{channel_id}/messages/{message_id}/replies**

Hugo Durand-mermet Hier 10:50
Question pour l'exo3, du coup pour chaque valeur négative ca tombera dans la category "transport" et positive dans "salary", c'est bien ca ?

▼ Réduire tout

Yoann Long Hier 10:51   👍 2
non pas forcément. La catégorie est donné dans le json pour chaque transaction

Hugo Durand-mermet Hier 10:52
ok, merci!

↩ Répondre

Exercise 04

Export all data into a SQL database. You must use **SQLite** as databse and **SqlAlchemy** as interface from python.

You should have :
- a table **teams** with teams_id and display_name
- a table **channels** with channel_id, display_name and a foreign key team_id
- a table **messages** with message_id, content, and a foreign key channel_id
- a table **replies** with reply_id, content, and a foreign key message_id

Raw SQL query such as "SELECT * FROM" **is forbiden**. You must use SqlAlchemy methods.

> You can use SQL GUI such as DataGrip from Jet Brains or DB Browser for SQLite **only as verification tools**